Distributed by:

JAMECO® ELECTRONICS

www.Jameco.com ✦ 1-800-831-4242

The content and copyrights of the attached material are the property of its owner.

599 Menlo Drive, Suite 100
Rocklin, California 95765, USA
Office: (916) 624-8333
Fax: (916) 624-8003

General: info@parallaxinc.com
Technical: stamptech@parallaxinc.com
Web Site: www.parallaxinc.com
Educational: www.stampsinclass.com

BS2p24 Demo Board (#45183)
BASIC Stamp 2p24 Demonstration Platform

Introduction

The BS2p Demo Board is designed as a low-cost platform to test and experiment with the new features of the BASIC Stamp 2p24, in particular: Dallas Semiconductor 1-Wire® communications, Philips I²C™ communications and working with standard parallel LCDs (with Hitachi HD44780 or compatible driver). The BS2p Demo Board is not exclusive to the BS2p; it is physically and electrically compatible with all 24-pin BS2-series BASIC Stamps.

The purpose of this document is to introduce the BS2p Demo Board and explain the particular configuration requirements for connecting Dallas 1-Wire® devices, Philips I2C™ devices and parallel LCDs.

Packing List

Verify that your BS2p Demo Board package is complete in accordance with the table below. For a selection of compatible 1-Wire® and I²C™ parts, order the BS2p "Plus-Pack" (#45184). The contents of the BS2p Demo Board package include:

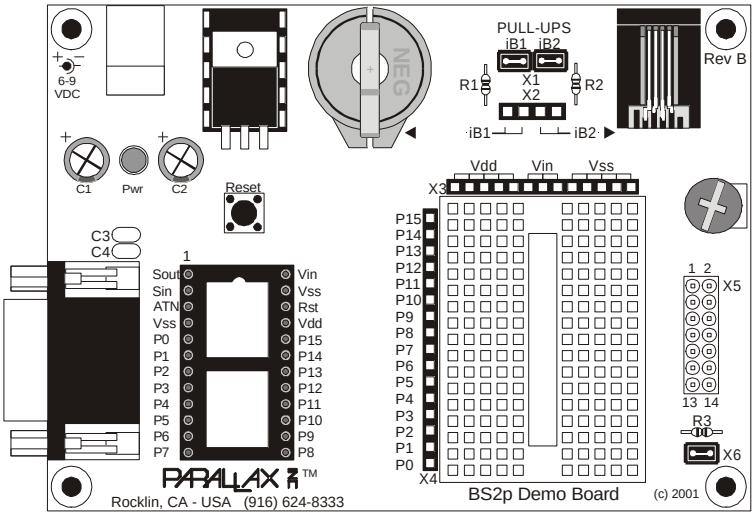
- Documentation
- BS2p24 Demo Board
- Jumper wires
- Source code on disk or CD-ROM

Features

A diagram of this board is shown on the following page.

- Standard 2.1 mm center-positive DC power jack
- DB9-F connector for serial programming
- On-board 5-volt regulator (LM2940 low-dropout)
- Solderless breadboard
- iButton™ socket (F5 MicroCan)
- RJ-11 jack for Dallas "Blue Dot Receptor" sockets
- 14-pin IDC connector for LCD (Parallax stocks a 2 x 16 LCD with cable - #27908)
- Contrast pot for LCD
- Selectable 4.7 k Ohm pull-ups (1-Wire® bus) and pull-down (LCD)
- Connections for P0 – P15, Vin, Vdd, Vss and 1-Wire® sockets

BS2p24 Demo Board Diagram



Connections and Jumpers

The BS2p24-IC's 16 I/O lines are accessible through the X4 socket at the left edge of the breadboard. This socket, and the others on the board, are compatible with 22-guage hookup wire and Parallax flexible jumpers (stock #800-00016). Vin (unregulated input), Vdd (regulated 5 volts) and Vss (ground) are made available through socket X3 at the top edge of the breadboard.

Connections to the 1-Wire® device sockets (iB1 and iB2) are made available through socket X2. The 4.7 kOhm pull-up resistor for the iButton socket is enabled by installing jumper iB1. The pull-up resistor for the RJ-11 "Blue Do Receptor" socket is enabled by installing jumper iB2. To use the iButton™ socket and the "Blue Dot Receptor" socket on the same BS2p I/O pin, install jumper iB1 or iB2, but not both (The iB1 and iB2 pull-ups may also be used for I²C™ components when no 1-Wire® devices are installed).

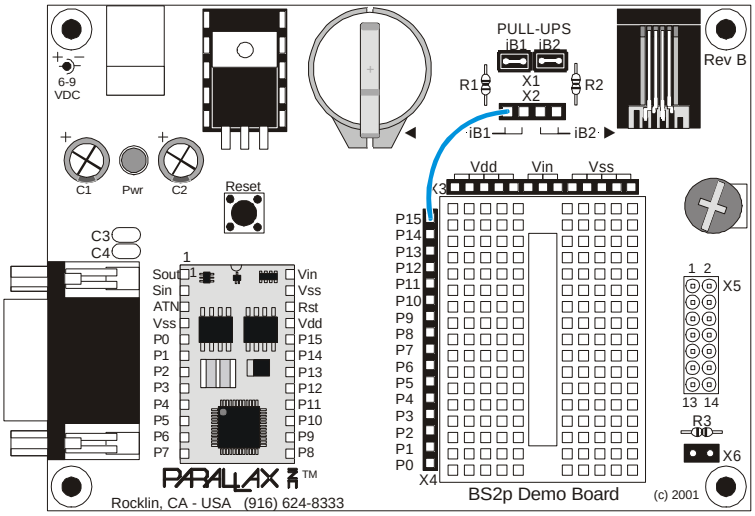
When using a parallel LCD with the demo board, install jumper X6 to enable the 4.7K pull-down resistor on the LCD's Enable (E) pin.

Experimenting With Dallas 1-Wire® Components

The BASIC Stamp 2p makes interfacing to Dallas 1-Wire® components very easy with the **OWOUT** and **OWIN** commands. The purpose of this experiment is to demonstrate these new commands by reading the unique ROM code from a 1-Wire® (iButton™) device. When using 1-Wire® components, the bus pin must be pulled-up to Vdd (5 volts) through a 4.7 kOhm resistor. The components for this experiment are included in the BS2p "Plus Pack."

Instructions for using 1-Wire components on BS2p Demo Board:

- Install BS2p in socket (note polarity indictors on BS2p and socket)
- Insert iButton™ into socket (markings down)
- Connect P15 to iB1 with 22-guage wire
- Enable iButton™ socket by installing pull-up jumper iB1
- Connect serial cable and 9-volt power supply
- Enter and run the program




```
' -----[ Title ]-----------------------------------------------------------
'
' File..... 1WIRE-ID.BSP
' Purpose... Reads ROM data from 1-Wire device
' Author.... Parallax
' E-mail.... stamptech@parallaxinc.com
' Started...
' Updated...

' {$STAMP BS2p}

' -----[ Program Description ]---------------------------------------------
'
' Reads 1-Wire device ROM pattern.  Data is displayed in DEBUG window

' -----[ I/O Definitions ]-------------------------------------------------
'
OWpin        CON    15                          ' 1-wire device pin

' -----[ Constants ]-------------------------------------------------------
'
OW_FERst     CON    %0001                       ' Front-End Reset
OW_BERst     CON    %0010                       ' Back-End Reset
OW_BitMode   CON    %0100
OW_HighSpd   CON    %1000
ReadROM      CON    $33                         ' read ID, serial num, CRC
SearchROM    CON    $F0                         ' search

NoDevice     CON    %11                         ' no device present

' -----[ Variables ]-------------------------------------------------------
'
idx          VAR    Byte                        ' loop counter
romData      VAR    Byte(8)                     ' ROM data from DS1820
devCheck     VAR    Nib                         ' device check return ocde

' -----[ Initialization ]--------------------------------------------------
'
Initialize:
  PAUSE 1000                                    ' let DEBUG window open

' -----[ Main Code ]-------------------------------------------------------
```

```
'
Main:
  DEBUG CLS
  GOSUB DeviceCheck                             ' look for device
  IF (devCheck <> NoDevice) THEN DisplayROM

NoDeviceFound:
  DEBUG "No 1-Wire device(s) present."
  END

DisplayROM:
  DEBUG "Dallas 1-Wire ID : "
  OWOUT OWpin,OW_FERst,[ReadROM]                ' send Read ROM command
  OWIN  OWpin,OW_BERst,[STR romData\8]          ' read serial number & CRC

  FOR idx = 0 TO 7
    DEBUG HEX2 romData(idx)," "                 ' show ID, serial num, CRC
  NEXT

  END

' -----[ Subroutines ]------------------------------------------------------
'
' This subroutine checks to see if any 1-Wire devices are present on the
' bus.  It does NOT search for ROM codes
'
DeviceCheck:
  devCheck = 0
  OWOUT OWpin,OW_FERst,[SearchROM]              ' reset and start search
  OWIN  OWpin,OW_BitMode,[devCheck.Bit1,devCheck.Bit0]
  RETURN
```

Experimenting With Philips I²C™ Components and Parallel LCDs

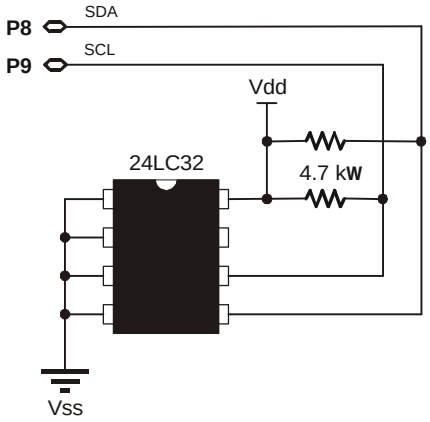
There are myriad components that communicate via the Philips I²C™ bus and the BASIC Stamp 2p makes connecting to these parts very straightforward with **I2COUT** and **I2CIN**. When using I²C™ parts, the SDA and SCL lines must be pulled up to Vdd (5 volts). The BS2p limits the SDA and SCL connections to pins 0 and 1 or 8 and 9, respectively. Refer to the Parallax Stamp Manual, version 2.0 (or later) for details.

Parallel LCDs, especially those controlled by the Hitachi HD44780 (or compatible) driver, have become as commonplace as LEDs in embedded applications. The BS2p supports the use of parallel LCDs with **LCDCMD**, **LCDOUT** and **LCDIN**.

This experiment demonstrates the use of the I²C™ protocol along with text and numeric output with an LCD. Specifically, the program generates an address and a pseudo-random value. The value is written to a Microchip 24LC32 EEPROM and read back. The address, output value and input value are displayed on the LCD. The LCD initialization includes the creation of a custom character (superscripted “2”).

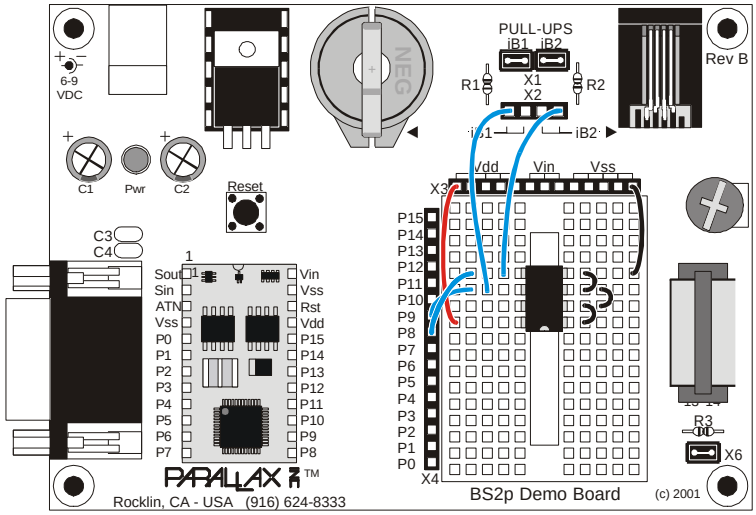
The schematic below shows the connections for the 24LC32 EEPROM. The LCD is connected to the lower eight I/O pins (OutL) of the BS2p24. To use a parallel LCD with the BS2p Demo Board, jumper X6 must be installed to enable the 4.7K pull-down resistor on the LCD’s Enable (E) pin.

Note: The components for this experiment are included in the BS2p “Plus Pack.”



Instructions for using I2C and LCDs with the BS2p24 Demo Board

- Install BS2p in socket (note polarity indictors on BS2p and socket)
- Remove all 1-Wire® parts
- Wire the 24LC32 in accordance with the schematic
- Install jumpers iB1, iB2 and X6
- Connect the LCD to X5
- Connect serial cable and 9-volt power supply
- Enter and run the program
- Adjust contrast pot for best display




```
' -----[ Title ]---------------------------------------------------------
'
' File..... 24LC32.BSP
' Purpose... Demonstrates I2CIN, I2COUT and using an LCD
' Author.... Parallax
' E-mail.... stamptech@parallaxinc.com
' Started...
' Updated...

' {$STAMP BS2p}

' -----[ Program Description ]---------------------------------------------
'
' Writes to and reads from I2C EEPROM.  Data is displayed on a 2x16 LCD.
'
' Program requires a 2x16 LCD
'
'   - LCD.E  --> Pin0 (pulled down [to ground] through 4.7K)
'   - LCD.R/W--> Pin2
'   - LCD.RS --> Pin3
'   - LCD.D4 --> Pin4
'   - LCD.D5 --> Pin5
'   - LCD.D6 --> Pin6
```

```
'   - LCD.D7  --> Pin7

' -----[ I/O Definitions ]-------------------------------------------------
'
LCDpin       CON    0                       ' LCD is connected to OutL
I2Cpin       CON    8                       ' SDA on 8; SCL on 9

' -----[ Constants ]-------------------------------------------------------
'
NoCmd        CON    0
ClrLCD       CON    $01                     ' clear the LCD
CrsrHm       CON    $02                     ' move cursor to home position
CrsrLf       CON    $10                     ' move cursor left
CrsrRt       CON    $14                     ' move cursor right
DispLf       CON    $18                     ' shift displayed chars left
DispRt       CON    $1C                     ' shift displayed chars right

DDRam        CON    $80                     ' Display Data RAM control
CGRam        CON    $40                     ' Custom character RAM control
Line1        CON    $80
Line2        CON    $C0

MaxEE        CON    4095                    ' highest EE address

' -----[ Variables ]-------------------------------------------------------
'
addr         VAR    Word                    ' EE address
addrHi       VAR    addr.HighByte
addrLo       VAR    addr.LowByte
rVar         VAR    Word                    ' for random number
tOut         VAR    Byte                    ' test value to LCD
tIn          VAR    Byte                    ' test value read from LCD
temp         VAR    Word                    ' temp value for display
width        VAR    Nib                     ' width of rt justified
pos          VAR    Byte                    ' column position
digits       VAR    Nib                     ' digits to display

' -----[ EEPROM Data ]-----------------------------------------------------
'
Super2       DATA   %01100                  ' superscript 2 (custom char)
             DATA   %00010
             DATA   %00100
             DATA   %01000
             DATA   %01110
             DATA   %00000
             DATA   %00000
             DATA   %00000

' -----[ Initialization ]--------------------------------------------------
'
LCD_Setup:
  PAUSE 500
  LCDCMD LCDpin,%00110000 : PAUSE 5      ' 8-bit mode
  LCDCMD LCDpin,%00110000 : PAUSE 0
  LCDCMD LCDpin,%00110000 : PAUSE 0
  LCDCMD LCDpin,%00100000                ' 4-bit mode
  LCDCMD LCDpin,%00101000                ' 2-line mode
  LCDCMD LCDpin,%00001100                ' no crsr, no blink
  LCDCMD LCDpin,%00000110                ' inc crsr, no disp shift

  ' download custom character map to LCD

  LCDCMD LCDpin,CGRam                    ' write to CGRAM
```

```
  FOR addr = Super2 TO (Super2 + 7)           ' build custom char
    READ addr,temp                            ' get byte from EEPROM
    LCDOUT LCDpin,NoCmd,[temp]                ' put into LCD CG RAM
  NEXT

' -----[ Main Code ]------------------------------------------------------
'
Splash:
  LCDOUT LCDpin,ClrLCD,["  BS2P <-> I",0,"C"]
  LCDOUT LCDpin,Line2, [" Communications"]
  PAUSE 2000

Main:
  LCDOUT LCDpin,ClrLCD,["I",0,"C:     Out="]
  LCDOUT LCDpin,Line2 + 10,["In="]

  FOR addr = 0 TO MaxEE STEP 5                ' create addresses
    RANDOM rVar                               ' create "random" value
    tOut = rVar.HighByte

    ' write value then read it back

    I2COUT I2Cpin,$A0,addrHi\addrLo,[tOut]
    PAUSE 100
    I2CIN I2Cpin,$A1,addrHi\addrLo,[tIn]

    ' display results

    LCDOUT LCDpin,Line1 + 4,[DEC addr]
    temp = tOut : width = 3 : pos = Line1 + 13
    GOSUB RJ_Print
    temp = tIn : width = 3 : pos = Line2 + 13
    GOSUB RJ_Print

    PAUSE 250
  NEXT

  END

' -----[ Subroutines ]-----------------------------------------------------
'
RJ_Print:                                     ' right justified printing
  digits = width
  LOOKDOWN temp,<[0,10,100,1000,65535],digits
  LCDOUT LCDpin,pos,[REP " "\(width-digits),DEC temp]
  RETURN
```

Additional Experiments

For more experiments with the BS2p and BS2p demo board, you may download the BS2p “Plus Pack” documentation and code samples from Parallax, free of charge.

Using Other BS2 Microcontrollers

The BS2p Demo Board is electrically compatible with the BS2, BS2e and BS2sx microcontrollers. The demo board is particularly useful for experiments with parallel LCDs (download the StampWorks experiments from Parallax for LCD examples using the BS2, BS2e and BS2sx). Please note that 1-Wire® interfacing is not possible with the BS2, BS2e and BS2sx without an external co-processor. I²C™ communications with these microcontrollers is possible with PBASIC code.